Privacy-aware Biometrics:
Design and Implementation of a Multimodal Verification System

Stelvio Cimato, Marco Gamassi, Vincenzo Piuri,
Roberto Sassi and Fabio Scotti
Dipartimento di Tecnologie dell'Informazione,
Università degli Studi di Milano, Via Bramante 65, 26013 Crema, Italy
e–mail: {cimato, gamassi, piuri, sassi, fscotti}@dti.unimi.it

Abstract

A serious concern in the design and use of biometric authentication systems is the privacy protection of the information derived from human biometric traits, especially since such traits cannot be replaced. Combining cryptography and biometrics, several recent works proposed to build the protection in the biometric templates themselves. While these solutions can increase the confidence in biometric systems when biometric information is stored for verification, they have been shown difficult to apply to real biometrics. In this work we present a biometric authentication technique that exploits multiple biometric traits. It is privacy-aware as it ensures privacy protection and allows the extraction of secure identifiers by means of cryptographic primitives. We also discuss the implementation of our approach by considering, as a significant example, the combination of iris and fingerprint biometrics and present experimental results obtained from real data. The implementation shows the feasibility of the scheme in practical applications.

1. Introduction

Biometric techniques are more and more deployed in several commercial, institutional, and forensic applications to build secure and accurate user authentication procedures. The interest in biometric approaches for authentication is increasing for their advantages such as security, accuracy, reliability, usability, and friendliness. As a matter of fact, biometric traits (e.g., fingerprints, voice, face), being physically part of the owner, are always available to the user who is therefore not afraid of losing them. They are one of the oldest form of identification (e.g., signature on a contract). However, compared to passwords, biometric traits cannot be strictly considered as "secrets" since often they can be inadvertently disclosed: fingerprints are left on a myriad of objects such as doors' handles or elevator buttons; pictures of faces are easily obtained without the cooperation of the subjects. Moreover, if they are captured or if their digital representations are stolen, they cannot be simply replaced or modified in any way, as it can be done with passwords or tokens [24]. These aspects have limited so far the number of applications in which biometric authentication procedures were allowed by privacy agencies in several countries. In addition to this, users often perceive the potential threat to their privacy and this reduces the user acceptance of biometric systems, especially on a large scale.

In a typical biometric authentication system, trusted users provide the authentication party with a sample of a biometric trait (e.g., a fingerprint scan). A digital representation of the fingerprint is then stored by the party and compared at each subsequent authentication with new fingerprint scans. The party is then in charge of protecting the database where digital representations of fingerprints are stored. If an intruder gained access to the database, she could prepare fake fingerprints starting from each of the digital images. To limit such a possibility, images of biometric traits are not stored explicitly: only a mathematical description of them is stored (the parameters of a model or relevant features). Such a mathematical characterization is generally called *template* and the information contained in it is sufficient to complete the authentication process. Templates are obtained through *feature extraction* algorithms. Often the database is completely avoided and each user carries with her a token, digitally signed and encrypted, where her template is stored. While such solutions are sensible and currently deployed, they are still critical from a privacy point of view since the biometric templates are exposed at risk of being decrypted and abused if the cryptographic keys are lost or stolen or the database protection violated.

In the literature, various strategies have been presented to address the problem of supporting personal verification based on human biometric traits, while ensuring a further level of protection (privacy) of digital templates [27]. Most approaches rely on jointly exploiting the characteristics of biometrics and cryptography [16, 13]. The main idea is that of devising biometric templates and authentication procedures which do not disclose any information on the original biometric traits, for example replicating the usual approach adopted in password-based authentication system. There, only a hashed version of the password is stored and the authentication procedure is carried on only comparing two hashes, the one stored and the other obtained from the newly typed password. In this way, the original password is never recovered (nor it might be) from its hashed version. Similarly, biometric templates are generated by using suited cryptographic primitives so as to protect their privacy and ensure that an attacker cannot retrieve any information on the original biometric trait used for the generation of the template. In this way, users' privacy is guaranteed. Moreover, even if a template is compromised (stolen, copied, etc.) it is always possible to generate a novel template by starting from the same original biometric trait. Biometric systems which guarantee this further level of protection might be termed *privacy-aware*.

The use of cryptographic primitives to protect biometric templates in privacy-aware systems poses a number of challenges. Different readings of the same biometric trait of the same individual, even if obtained by using the same sensor in a short period of time, always show some variability. For this reason they cannot be directly exploited to secure the biometric templates by means of standard cryptographic techniques. In these techniques, cryptographic keys have zero uncertainty and a single-bit difference (in the key or in the encrypted data) spoils the possibility of accessing the original data. The use of biometrics as cryptographic keys for protecting the biometrics traits should therefore be *error tolerant*, since biometric readings are always different: generating cryptographic keys from biometrics relies on an error tolerant binary representation of the biometric features [14]. A comprehensive survey of different approaches presented in the literature and the related limits can be found in [27]. Biohashing and its variants have been presented in [20] as a solution in which a biometric template is randomized by using a pseudo-random token. However, the security of such approaches is broken if the pseudo-random token is stolen or copied. Other variants have been proposed to face this problem [19].

In this paper, we propose a privacy-aware biometric cryptographic scheme which, building over previous works, enables the creation of a unique identifier associated with each enrolled person by exploiting the error tolerant properties of the biometric templates. This is obtained by using multiple biometric traits concurrently and the recently introduced cryptographic primitives secure sketches and fuzzy extractors. The resulting scheme is *multimodal*, in the sense that multiple biometric traits (at least two) can be used. Other proposals based on secure sketches have been presented; however, they have been shown difficult to apply to real biometrics and the construction of practical systems still is an open issue [26, 4]. In general, we feel that the main aspect which has not been sufficiently studied is the optimum use of the design opportunities offered by biometric multimodal systems. The contribution of this work is threefold. First, we identify the requirements that a privacy-aware multimodal biometric system should satisfy. Second, we propose such a privacy-aware system to provide an effective and easily deployable identity verification system. Third, we suggest a practical implementation of our method based on real biometrics.

The outline of the work is as follows. Section 2 discusses approaches presented in the literature. In Section 3 we sketch the main characteristics that a biometric system should present to overcome privacy related issues. In Section 4, we present the design methodology suited to create privacy-aware biometric verification systems with the desired degree of security and privacy protection. The basic components and the (parallel and hierarchical) compositions according which they can be arranged are also introduced. In Section 5, we then describe an actual implementation of the scheme. Given the fact that the construction of practical systems is critical and many issues indeed relate to implementation, the section enriches the description of the scheme. We also report experimental data obtained from real biometrical datasets. Finally, we give our conclusions in Section 6.

2. Related work

Several biometric authentication techniques, based on the use of error correcting codes (ECCs) to cope with the variability of biometric templates, have been presented in literature. Juels and Wattenberg [16] proposed the *fuzzy commitment* scheme, where a secret message is protected by using a biometric template. In this case, an error correcting code is used to associate a codeword c with a person and compute an offset ($\delta = c \oplus x$) for the biometric template x. The encrypted message (the fuzzy commitment) is then represented by the pair $\{\delta, h(c)\}$, where $h(c)$ is a one-way hash function. Moving in the same direction, Hao et al. [13] proposed a biometric key generation procedure, based on an iris code feature extraction algorithm and on the combined use of Hadamard and Reed-Solomon codes. Juels and Sudan [15] also proposed a *fuzzy vault scheme* relying on the polynomial interpolation technique to cope with variability of the stored biometric templates. Recently, a similar approach has been proposed in [25] to achieve a biometric system for offline verification of certified, cryptographically secure documents. The presented technique

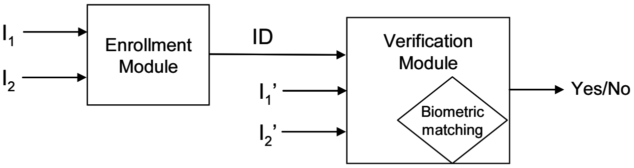

Figure 1. The overall structure of the multimodal biometric authentication system.

can produce printable IDs obtained from an extracted and compressed iris feature and an arbitrary text.

The problem of generating strong keys from biometric readings has been addressed by Dodis et al [8], where the properties of both secure sketches and fuzzy extractors primitives have been analyzed. In [1], the author points out how the multiple use of the same fuzzy secret can cause security problems, and can introduce outsider and insider attack scenarios, where an adversary tries to obtain information on the secret by performing repeatedly extractions and regenerations of the fuzzy secret. In such scenarios, with some limitations, it is possible to show that information theoretic security can be achieved and existing constructions can be adapted to satisfy the additional requirements. More general attack models and constructions to achieve a secure remote biometric authentication are proposed in [2]. A general framework to design and analyze a secure sketch for biometric templates is presented in [26], where face biometrics have been used as case study. Interestingly, the paper shows that theoretical bounds have their limitations in practical schemes. In particular, it has been shown that the entropy loss of the template cannot be considered a complete description of the robustness level of the scheme in practical applications, while the analysis of the *false match rate* (FMR, i.e., the probability of an individual not enrolled being identified) and *false non-match rate* (FNMR, i.e., the probability of an enrolled individual not being identified by the system) should be always envisioned. Finally, the application of a fuzzy sketch based scheme to iris biometrics has been presented in [3]. The paper relies on a near-optimal error-correcting code (based on a two-dimensional iterative min-sum decoding algorithm) and provides also an explicit estimation of the upper bounds on the correction capacity of such a kind of schemes.

3. Requirements

A first step in the construction of a privacy-aware multimodal biometric system is the identification of the requirements it should have. In particular, we have identified the following requirements.

1. *Privacy-awareness*. The system should be able to build user identifiers or templates from which it should be practically impossible to recover a representation of the actual biometric traits. For doing so it can employ an efficient encryption scheme that converts noisy nonuniform inputs (like biometric readings are) in easily and reliably reproducible binary strings with a certain degree of tolerance in the given inputs. Privacy-awareness might reduce the perceived threat to privacy and could overcome the legal issues related to the respect of privacy protection laws, currently ruling in several countries.

2. *Multi-modality*. Multiple readings of the same biometric trait (e.g., the fingerprint of different fingers or the iris of the two eyes) or multiple different traits should be considered. Multimodal systems are know to display a higher reliability [23] and this might increase user acceptance in a wider spectrum of applications. Moreover, given a certain level of privacy protection, the trust in the authentication procedure should scale with the number of traits (e.g., admission to critical areas could require a larger number of traits to be verified).

3. *Modularity*. The design should be modular with respect to the basic biometric encryption modules. A larger number of biometric traits should be added by simply composing the basic modules. Besides simplifying the design process, this allows for a tuning of the structure of the system to the privacy protection degree requested by the application, thus offering different levels of security in authentication at appropriate costs.

4. *Independence*. The overall scheme of the system should be independent from the biometric traits selected and from specific feature extraction algorithms implementing proprietary solutions. Besides, as soon as available, more accurate techniques for biometric recognition (e.g., with improved error rates or relying on novel traits) can be directly and easily incorporated in the biometric authentication system. This allows for updating continuously the global solution by exploiting the opportunities offered by the state of the art. On the other hand, since the biometric system for a specific application is realized by combining components based on well-known algorithms and its characteristics can be directly derived from the ones of these components, the resulting system will be easy to understand and be accepted by the application owner.

5. *Independence from a centralized repository of identities*. The system should not rely on the availability of a central database supporting the authentication procedure. National privacy agencies often rule against such

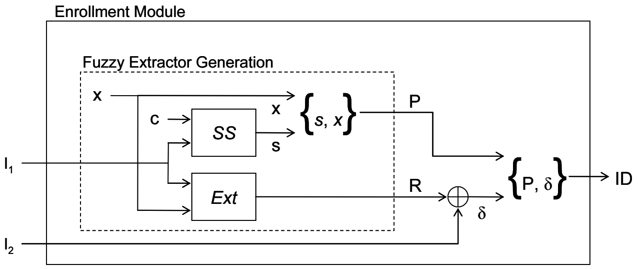

Figure 2. The Basic Enrollment Module.

databases. Also, the system should not rely on network architecture for cryptographic authentication to reduce the points of failure.

6. *Deployability*. The system should be deployable. The overall encryption and processing schemes should be computationally efficient enough to be implemented also in real-time applications. The overall structure should be compact and configurable so as to be easily tailored to the real needs of the applications.

4. General Scheme

In this section we describe the scheme of our multimodal biometric system. We consider first that only two biometric traits are employed concurrently; extensions to a larger number of biometric traits will be obtained by the composition of basic modules (see Section 4.3). For each biometric trait a feature extraction algorithm $\mathcal{F}_i$ is selected among the ones available in the literature. The algorithm, given a digital representation of the trait, generates a mathematical description that can then be turned in a digital string I_i (n_i bit long). In the following, to simplify the discussion we refer to I_i as *biometric input*. We assume that for at least one of the two feature extraction algorithms, it is possible to measure its error rate e_i (i.e., the rate of bits in the pattern I_i which could be modified without affecting the biometric verification of the subject). Without loss of generality, we denote such an algorithm as $\mathcal{F}_1$.

With regards to inputs and outputs, the overall scheme resembles a common multimodal system and is depicted in Figure 1. It is composed of two basic modules: the *enrollment* module creates an ID starting from the biometric readings of a user. The ID can be envisioned as a function of the binary strings I_1 and I_2 and is associated with the owner of the biometric traits. The ID is then stored or printed on a document and must be provided during the verification phase. The *verification* module verifies the identity claimed by the user using the ID and novel biometric readings (biometric inputs I'_1 and I'_2). The process is successful if the novel readings match the ones used to build the ID.

4.1. Preliminaries: the fuzzy extractor primitive

As briefly stated in the introduction, one of the problems in deriving cryptographic keys from biometric traits is that digital representations of the same biometric trait always differ slightly. The same sort of differences are encountered also among templates. Obviously, a single-bit difference in a binary string (e.g., a password), by construction, makes it impossible to recover the secret or validate an authentication procedure. The first problem that needs to be solved is therefore the one of obtaining reliably reproducible binary strings from noisy non-uniform inputs.

The *secure* or *fuzzy sketch* [8] is a cryptographic primitive that solves the problem of error tolerance. It enables the computation of a public string P from a binary string r such that from another binary string r' sufficiently close to r it is possible to reconstruct the original one. In this construction, the knowledge of P (which is made public), does not reveal enough information on the original secret reading r, provided that the entropy of r is large enough. Secure sketches are therefore attractive in the context of biometrics, given the large entropy of biometric templates. Unfortunately, generally speaking, entropy is not uniformly distributed along biometric templates and low entropy regions do exist. Among other reasons, this might be easily understood considering that templates usually are formatted according to international standards (e.g., ANSI INCITS 378-2004 for fingerprints) and then follow a regular structure. Moving a step further, *fuzzy extractors* [1] address the problem of non-uniformity by associating a random uniform string R to the public string P still preserving the error-tolerance property of fuzzy sketches. Indeed, fuzzy extractors can be constructed from fuzzy sketches and enable the recovering of the secret uniform random string R, from the knowledge of the public string P and a reading r' sufficiently close to r. A fuzzy extractor can be seen as pair of functions: *Generate* (*Gen*) and *Reproduce* (*Rep*). *Gen* is a randomized generation function that from the input binary string w produces a private binary string R and a public binary string P. The construction guarantees that the probability density function of the bits in R is close to uniform even for those who observe P. *Rep* is a regeneration function that, given in input a public string P obtained from the *Gen* procedure and a value w' close enough to w with respect to a certain metric, returns a string S such that $S = R$.

The application of a fuzzy extractor to biometric templates in the real world poses a number of problems. Biometric templates have different formats, which are not always compatible with the application of fuzzy extractors, and the definition of a distance metric among templates is not always straightforward. Furthermore, at the core of fuzzy extractors typically lies an error correcting code. The variability among different readings of the same biometric trait is often larger than the correction capabilities of most codes and special constructions are needed. (More details on the specific fuzzy-extractor we used are reported in the next section and some practical details in Section 5).

4.2. The Basic Modules

A simplified sketch of the *enrollment* phase is reported in Figure 2 where the basic *enrollment module* is depicted. A novel identifier ID is created for each user, by composing the available biometric features. The first biometric input I_1 is used as input to the generation function of a fuzzy extractor that returns a public string P, and a secret R. The secret string R is then xor-ed with I_2 to produce the resulting binary string δ, that together with P constitute the ID for the user. The construction guarantees that the randomness in R is uniformly distributed, therefore from the ID it is not possible to reconstruct I_2. The strings P and R are produced directly by the Gen procedure of the fuzzy extractor which has been built out of a secure sketch SS, according to the construction proposed in [1]. The secret uniform random string R is computed as $R = Ext(I_1, x)$, where $Ext(w; x)$ is the application of a strong extractor with randomness x . A possible strong extractor is constructed selecting a random binary string x and using it as key in a Hash-based Message Authentication Code (HMAC). The public string P is computed as $P = SS(I_1; c)||x$, where $SS(w; c)$ is the output of the secure sketch with randomness c, used in the construction of the fuzzy extractor. In practise, one selects an error correcting code with n_1 bits-long codewords and error correcting capability $t = e_1 \times n_1$. Then, a random codeword c is selected and the distance between c and I_1 is computed as $s = I_1 \oplus c$.

The *verification module*, illustrate in Figure 3, combines the ID associated with the user and two fresh biometric readings to execute the authentication procedure through biometric matching. The digital representations of the biometric traits are processed through the same algorithms selected for enrollment (e.g., $\mathcal{F}_1$ and $\mathcal{F}_2$) leading to the binary strings I'_1 and I'_2. Given the variability inherent to biometrics, I_1 and I_2 are similar to I'_1 and I'_2 respectively, with respect to a certain metric. The verification module relies on the regeneration phase of the fuzzy extractor, which employing I'_1 and the public string $P = \{s, x\}$ regenerates the same secret string R obtained from I_1. More in detail, $c' = I'_1 \oplus s$ is a corrupted version of c, if the fresh reading I'_1 is sufficiently close to the enrolled feature I_1. In this case the Rec phase of the secure sketch embedded in the fuzzy extractor will return the string I_1. In fact, processing c' with the decoding algorithm of the selected error correcting code one might obtain c which in turn leads to $I_1 = c \oplus s$. With I_1 in hand, R is obtained following the same path used at enrollment: I_1 is given in input to the strong extractor Ext together with x contained in the public string P. Finally, the reconstruction of the second biometric feature I_2 is obtained from R as $I_2 = R \oplus \delta$. The verification succeeds if the biometric matching between I_2 and the I'_2 is positive. It is worth noticing that differently from other approaches that are based on fuzzy sketches or extractors, the verification phase relies on a biometric matcher and not on a direct comparison between reconstructed strings. If more accurate matching modules were developed for the same biometric trait, it would be possible to embed them into the scheme with no impact on the remaining modules. Moreover, notice that no requirements are set for the construction of the matcher.

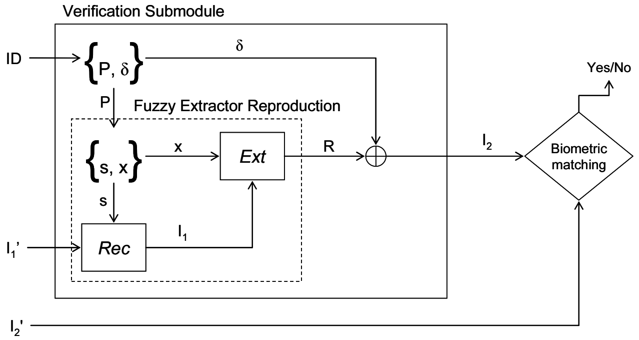

Figure 3. The Basic Verification Module.

4.3. Composition of basic modules

The composition of the basic modules enables the creation of authentication applications having different levels of security and using a higher number of biometric features. The basic enrollment and verification modules can be combined hierarchically and/or in parallel (with respect to the input biometric readings). Figure 4 shows the layout of the described compositions.

The *parallel composition* (Figure 4(A)) offers a simple method to exploit different biometric traits to create the ID. This way, the level of multi-modality implemented is higher than in the basic approach since more than two biometric traits are in use. Given a certain number of biometric traits, the corresponding binary strings J_i are obtained from the digital representations of the traits. The two inputs I_1 and I_2 to the enrollment module described in Section 4.2 are obtained through the concatenation of strings J_i. In particular, I_1 is obtained from $\{J_1, J_2, \ldots, J_k\}$ and I_2 from $\{J_{k+1}, J_{k+2}, \ldots, J_N\}$ where N is the number of different biometrics. Analogously to what required for the basic module, it should be possible to measure the error rates e_i for all the feature extraction algorithms that generated J_i

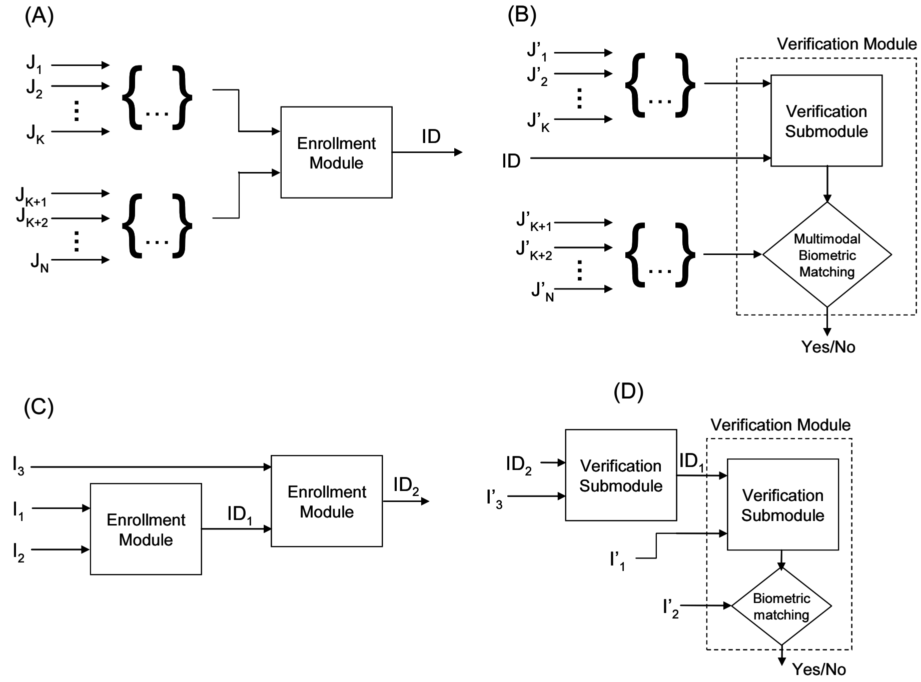

Figure 4. Examples of the enrollment and verification modules in a parallel composition (A)-(B) and in a hierarchical composition(C)-(D).

with $i \in [1, k]$. However, the global error rate e in the composed input I_1 needs particular scrutiny as each biometric method differently contributes to the overall error rate. Notice with respect to Figure 4(B) that in the verification phase the biometric matching module is truly multimodal, that is, it receives in input a composition of $N - k$ biometric readings $(J_{k+1}, J_{k+2}, \ldots, J_N)$ to be matched against the ones collected at enrollment.

The basic modules can be composed also in *hierarchical* structures. Figures 4(C) and 4(D) show an example of a two-level hierarchical composition. Biometric inputs I_1 and I_2 are used to create ID_1 by means of a basic enrollment module. Then, ID_1 is used in place of the second biometric trait in a cascaded basic enrollment module together with a third biometric input I_3. The binary string ID_2 is finally associated with the user. In the verification phase, ID_2 and a binary string I'_3 obtained from a fresh biometric reading are processed through a first verification submodule (substantially a fuzzy extractor; see Figure (3)) and ID_1 is recovered. Finally, a basic verification module receives ID_1, I'_2, and I'_1 as input and completes the authentication process.

It is worth noticing that it is possible to build more complex systems by using each method of composition (parallel and hierarchical) recursively or by combining the methods iteratively.

4.4. Analysis of the method

To foolish the authentication system, an adversary can: *i*) obtain the digital representations of the biometric traits of a genuine user through covert means; *ii*) or recover I_i from what is publicly available and associated with the enrolled person (the identifier). In the first case, to attack the system, the adversary should steal at least two biometric samples and compute I_i to complete successfully the authentication phase. As described in Section 4.3, a higher number of biometrics can be taken into account in the setup of the authentication system to increase the overall security of the application and prevent such kind of attacks. In the second case, the method should ensure that the adversary cannot take advantage from the knowledge of the identifiers or from tampering with the enrollment and verification procedures. Indeed, our approach builds on the fuzzy commitment scheme presented by Juels and Wattenberg and recast as secure sketch in [1, 7]. Differently from Juels's approach, in our scheme, we make use of a fuzzy extractor [8, 1] that guarantees both uniformity and error tolerance in reconstructing the biometric inputs I_1 and I_2. The assumption when using a fuzzy extractor is that the public information P must be sufficiently separate from the extracted secret R, so that P does not leak information on the biometric input I. Indeed, as shown in [10, 9], the mutual information between P and $w = I_1$ must be non trivial, that is, P must leak some information about the biometric input I_1 in order to correct errors in inputs similar to I_1, even if the input distribution is uniform. In this case, it is possible to use a weaker notion of security and to define *entropically secure* fuzzy extractors, that is, fuzzy extractors for which the knowledge of (R, P) does not help in predicting the value $f(I_1)$ for any predefined function $f(w)$. An equivalent definition is the one of *uniform fuzzy extractor*, that is, when the probability density function of R and P might be considered close to uniform. If the adversary has the capability to tamper the public string P returned by the fuzzy extractor, another abstraction *robust fuzzy extractors* can be considered. For this kind of extractors the retrieve procedure recovers the secret string R only if the original public string P is given as input; otherwise, a special symbol is produced. By using a robust uniform fuzzy extractor, the proposed scheme ensures both the randomness of R and the protection from adversarial attempts to use the information in P to recover the original biometric input readings.

In our scheme the second biometric reading is xor-ed with the resulting bit-string obtained after processing the first biometric reading, which is then used as a key. From the previous discussion, the randomness of the key is ensured by the fuzzy cryptographic primitive used in the enrollment phase. To have strong security guarantees, it should be also ensured that the biometric features extracted from the reading are not too much biased, avoiding that the adversary can collect information on the string used as key in the xor-ing. For this reason, the second biometric input should ensure a sufficiently large and uniform entropy.

5. Implementation and Experimental Results

Privacy-aware biometric systems while theoretically conceivable are often difficult to apply to real biometrics. For this reason, the implementation described in this section not only shows that the method described in Section 4 is practically feasible, but also casts light on the method itself.

Our implementation is based on two biometric traits: iris and fingerprint. Since the work of Daugman [6], binary strings (often called *iriscodes*) are obtained from pictures of the eye by using banks of Gabor's filters. Genuine subjects and impostors are then discriminated using the Hamming metric on such strings. Following the terminology used in this paper, iris codes correspond to binary input I_1 and the feature extraction algorithm employed to generate them correspond to $\mathcal{F}_1$. By using the code presented in [22], we were able to compute 9600 bits wide iris codes (radial resolution: 20). The code displays an error rate e_1 of about 40%. Fingerprints templates (I_2) were instead computed by using the NIST NBIS code `mindtct` [28] (feature extraction algorithm $\mathcal{F}_2$); the 34 best quality minutia were selected and then serialized in a ANSI INCITS 378-2004 record (1920 bits). The biometric match between fingerprint templates was verified by using the NIST NBIS matcher `bozorth3`. The matcher returns a similarity value between the two minutia sets; to obtain a Hamming distance, as suggested in the best practice of the literature of multimodal biometrics, the `bozorth3` score was subtracted from a large value (500) and then normalized in the range $[0, 1]$.

5.1. Construction of the fuzzy extractor

We have implemented the *Gen* procedure of the fuzzy extractor as follows. First, a 128 bit random number x was drawn and used as key in the HMAC-SHA1 algorithm (strong extractor *Ext*), as provided by the standard Java JDK, that processed I_1 to obtain the pseudo-random secret R. Since the number of bits in R must match the size of the biometric input I_2, which is a string of 1920-bit, we applied repeatedly (12 times) the HMAC-SHA1 algorithm (HMAC-SHA1 returns a string which is 160 bits long). Then, we selected a shortened Reed-Solomon $[9600, 1920, 7681]_{2^{14}}$ random codeword c [17]. The string $s = \tilde{I}_1 \oplus c$ is computed as the binary shift necessary to obtain c from $\tilde{I}_1$, where $\tilde{I}_1$ is the 9600 bit iris code preliminary mapped with a $[14, 1, 1]_2$ naive code. The mapping might be rationalized as follow. The codeword c is built with symbols that are 14 bits long. Each of the 9600 bits of the iris code is turned into a 14-bits symbol simply padding it with zeros, which is what the coding we selected does. Such a coding ensures that at most one bit in each symbols of c might be corrupted. One might wonder why we did not simply packed the bits together to form a series of m bits symbols as in common industrial application. The reason is that we want to correct *at most* a certain number of errors and not *at least*, as usual. The selection of a proper error correction code is critical and not trivial (see Appendix A for further discussion on this issue). Finally, x was concatenated with s to obtain the string P, which can be made public without impairing the security of the scheme.

Analogously, the reproduction function *Rep* was similarly built. In practice, one decomposes P into x and s and then applies the shift s to I'_1 to obtain a corrupted version of c. If the number of bits that differ from I_1 and I'_1 is smaller than $t = 3840$, the error correction capability of the Reed-Solomon code, the codeword can be decoded. The codeword c is obtained as $c = \mathrm{RSenc}(\mathrm{RSdec}(s \oplus I'_1))$, where RSend and RSdec are a pair of Reed-Solomon encoding and decoding algorithms. Then, $I_1 = s \oplus c$ furnished at enrollment is recovered. Analogously to what done in the *Gen* phase, I_1 is set as input to the strong extractor *Ext* with randomness x (HMAC-SHA1) to obtain R.

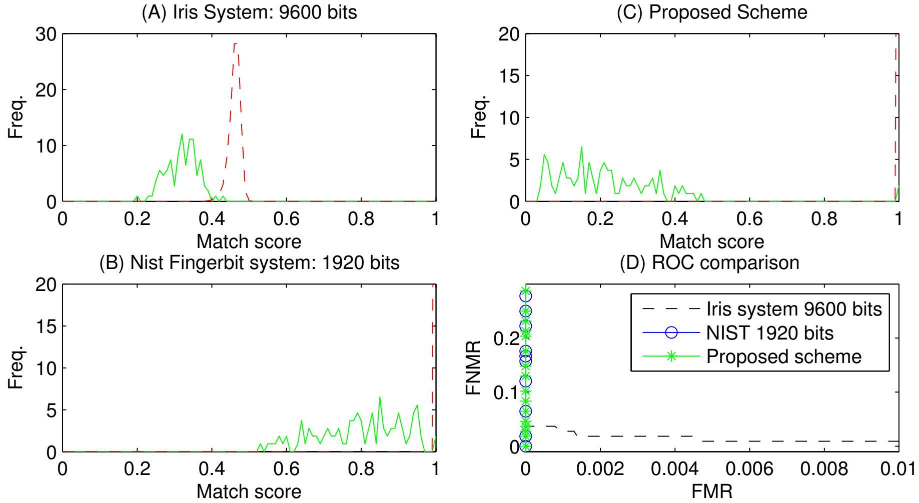

Figure 5. Frequency distributions and ROC curves for a practical implementation of the multimodal biometric authentication system (panels (C) and (D)). As a reference, in panel (A) and (B) we reported the frequency distributions of the single-trait biometric systems on which our implementation built (dashed-line: impostor). Correspondent ROC curves are included in panel (D).

5.2. Experimental Results

We made the assumption that the enrolling agency desires to collect only biometrics of sufficient quality and that more than one sample could be required for each subject to ensure such a quality. We further supposed that three different iris pictures and fingerprint scans should suffice; among the three iris codes computed we retained the one with the smallest number of masking bits[1] (I_1). For each fingerprint's minutia, `mindtct` offered a quality estimate; the fingerprints template with the highest average quality was further processed (I_2). We performed our experiments by coupling eyes images from the CASIA *iris database* [5] to fingerprints scans extracted from the FVC2000 dataset. In particular, we synthetically created a dataset of 108 individuals. For each individual, we had three eye and fingerprint images to be used in the enrollment phase, and four eye images and five fingerprint images for the verification phase [21].

At enrollment, I_1 was processed through the *Gen* phase of the fuzzy extractor to obtain R and P. Then, the offset $\delta = R \oplus I_2$ was concatenated with P to form the ID. The procedure was repeated for each of the 108 individuals.

For the verification phase, we quantified both the FNMR, by applying the basic verification module to biometric inputs collected from the same subject, and the FMR, by trying to validate the ID against all the other subjects. First, the ID was split into δ and P. The *Gen* phase of the fuzzy extractor ensures that as long as a second iris code I'_1 is close enough to the iris code collected at enrollment, the secret R might be obtained only from the knowledge of P. Obviously this condition should fail for an impostor. Therefore, when the decoding operation $\mathrm{RSdec}(s \oplus I'_1)$ failed using each of the four available iris codes in the validation set, the Hamming distance between the two subjects being verified was set to 1. Otherwise, with R and δ at hand, the fingerprint template might be retrieved, by computing $R \oplus \delta$. Then, once acquired a second fingerprint sample a biometric match could be performed, and its result determined the success (or not) of the verification procedure. The biometric match was performed with each of the five fingerprint images available.

We selected as references for a comparison the performances of the two biometric systems based only on iris or fingerprint, respectively. Such performances were evaluated on the same dataset and using an identical approach for enrollment and verification (*best-of-three* in enrollment; *best-of-four* in verification for the iris system and *best-of-five* for the fingerprint system). Figures 5(A)-(C) present the frequency distributions for different values of the match-threshold for the single-trait biometric system and for our method. Moreover, Receiver Operating Characteristic (ROC) curves are reported in Figure 5(D). The single iris system showed a Equal Error Rate (EER), (i.e., the value of the threshold used in the discriminating procedure at which FMR and FNMR are identical) of 0.9%, while the fingerprint system and the proposed scheme achieved a

[1] For each bit of the iris code, there is a correspondent masking bit that denotes its quality; a one masking bit means that the iris code in that position is affected by errors occurred in the segmentation procedure.

EER=0%. As expected for multimodal systems, the scheme we suggested, while improving the protection of the biometric inputs, showed a performance which is equivalent to the one of the single-trait fingerprint system (which is the best performer in our practical implementation).

By using commercial iris-code segmentation libraries, we are sure that better absolute rates could be obtained. Also, larger datasets could be employed to have more realistic estimates of the EER. However, the implementation of our method was developed mainly to verify the practical feasibility itself and it fulfills such goal.

6. Conclusions

In this paper, we have proposed a method combining standard cryptographic techniques and biometrics to provide an effective and easily deployable identity verification system. The system is privacy-aware since the information contained in the identifier is not sufficient to recover the biometric traits of the users and further biometric inputs are required. Any abuse of biometric information is then prevented. With respect to the requirements discussed in Section 3, it is easy to see that Requirement 1 was completely fulfilled. The method is multimodal (Requirement 2) needing at least two biometric traits. Moreover, the method is composed of two basic modules, that can then be combined to build more complex systems (Requirement 3) and it does not depend on the particular feature extraction algorithms selected (Requirement 4).

The method we propose enables the biometric verification of persons by using offline secure documents, in which neither biometrics traits nor other sensible data are stored in a central database (Requirement 5). To ensure its validity, the identifier produced during the enrollment phase could be signed using the private key of the issuer. Then, at verification, the signature on the ID could be verified using the issuer's public key.

Finally, we suggested an actual implementation of our method based on real biometrics. The implementation shows the feasibility of the scheme (Requirement 6) and offers an idea of the performances one might obtain from the application on real datasets. Indeed, the resulting error rate is acceptable and it is not worse than the best error rate of the single-trait biometric systems on which it is based. The work paves the way for large scale applicability of privacy-aware biometric systems.

7. Acknowledgments

The authors acknowledge helpful conversations with Sabrina De Capitani di Vimercati while writing the paper. The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n 216483.

References

[1] X. Boyen. Reusable cryptographic fuzzy extractors. In *Proc. of the 11th ACM Conference on Computer and Communication Security (CCS 2004)*, volume 3027, pages 82–91. ACM, 2004.

[2] X. Boyen, Y. Dodis, J. Katz, R. Ostrovsky, and A. Smith. Secure remote authentication using biometric data. In R. Cramer, editor, *Advances in Cryptology (EUROCRYPT 2005)*, volume 3494 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005.

[3] J. Bringer, H. Chabanne, G. Cohen, B. Kindari, and G. Zemor. An application of the goldwasser-micali cryptosystem to biometric authentication. In *Proc. of the 12th Australasian Conference on Information Security and Privacy (ACISP'07)*, volume 4586 of *Lecture Notes in Computer Science*, pages 96–106. Springer-Verlag, 2007.

[4] J. Bringer, H. Chabanne, G. Cohen, B. Kindarji, and G. Zémor. Optimal iris fuzzy sketches. *The Computing Research Repository*, abs/0705.3740, 2007.

[5] Chinese Academy of Sciences. Database of 756 greyscale eye images; Version 1.0, 2003.

[6] J. G. Daugman. High confidence visual recognition of persons by a test of statistical indenpendence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:1148–1161, 1993.

[7] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. Technical Report 2006/235, Cryptology Eprint Archive, 2006.

[8] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology (EUROCRYPT 2004)*, volume 3027 of *Lecture Notes in Computer Science*. Springer-Verlag, 2004.

[9] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors. In P. Tuyls and J. Goseling, editors, *Security with Noisy Data*, chapter 5, pages 93–111. Springer-Verlag, 2007.

[10] Y. Dodis and A. Smith. Correcting errors without leaking partial information. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 654–663, 2005.

[11] W. J. Gross, F. R. Kschischang, R. Koetter, and P. G. Gulak. Towards a VLSI architecture for interpolation-based soft-decision Reed-Solomon decoders. *The Journal of VLSI Signal Processing*, 39(1-2):93–111, 2005.

[12] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Trans. Inf. Theory*, 45(6):1757–1767, 1999.

[13] F. Hao, R. Anderson, and J. Daugman. Combining cryptography with biometrics effectively. Technical Report UCAM-CL-TR-640, University of Cambridge, Computer Laboratory, United Kingdom, July 2005.

[14] A. K. Jain, A. Ross, and S. Pankanti. Biometrics: A tool for information security. *IEEE transactions on information forensics and security*, 1(2):125–143, June 2006.

[15] A. Juels and M. Sudan. A fuzzy vault scheme. In A. Lapidoth and E. Teletar, editors, *Proceedings of the IEEE International Symposium on Information Theory, 2002*, page 408. IEEE Press, 2002.

[16] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *Proceedings of the 6th ACM conference on Computer and communications security (CCS '99)*, pages 28–36, New York, NY, USA, 1999. ACM Press.

[17] P. Karn. Reed-solomon encoding and decoding code, 2002.

[18] R. Koetter and A. Vardy. Algebraic soft-decision decoding of Reed-Solomon codes. *IEEE Trans. Inf. Theory*, 49(11):2809–2825, 2003.

[19] A. W.-K. Kong, K. H. Cheung, D. Zhang, M. S. Kamel, and J. You. An analysis of biohashing and its variants. *Pattern Recognition*, 39(7):1359–1368, 2006.

[20] A. Lumini and L. Nanni. An improved biohashing for human authentication. *Pattern Recognition*, 40(3):1057–1065, 2007.

[21] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain. FVC2000: Fingerprint verification competition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):402–412, 2002.

[22] L. Masek and P. Kovesi. MATLAB source code for a biometric identification system based on iris patterns. The School of Computer Science and Software Engineering, The University of Western Australia, 2003.

[23] A. Ross, K. Nandakumar, and A. K. Jain. *Handbook of Multibiometrics (International Series on Biometrics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[24] B. Schneier. Biometrics: uses and abuses. *Commun. ACM*, 42(8):136, Aug. 1999.

[25] D. Schonberg and D. Kirovski. Eyecerts. *IEEE Transactions on Information Forensics and Security*, 1:144–153, June 2006.

[26] Y. Sutcu, Q. Li, and N. Memon. Protecting biometric templates with sketch: Theory and practice. *IEEE Transaction on Information Forensics and Security*, 2(3), 2007.

[27] U. Uludag, S. Pankanti, S. Prabhakar, and A. Jain. Biometric cryptosystems: Issues and challenges. In *Proceedings of the IEEE, Special Issue on Enabling Security Technologies for Digital Rights Management*, volume 92, pages 948–960, June 2004.

[28] C. I. Watson, M. D. Garris, E. Tabassi, C. L. Wilson, R. M. McCabe, S. Janet, and K. Ko. User's Guide to NIST Biometric Image Software (NBIS). (formerly NISTIR 6813), 2007.

Appendices

A. A short discussion on the ECC code employed

The selection of the error correcting code needs further discussion. Given the large inter-subject variability of iris templates, for which typically $e_1 > 0.25$, the fraction of errors the code must be able to withstand is larger than in usual ECC applications. Common ECC code, like BCH, are capable of correcting a fraction of errors strictly less than $n/4$, thus seems ruled out. Others binary codes might get closer to the Singleton bound but at the price of a small rate k/n. In fact, as several authors pointed out [9], the Plotkin bound from coding theory implies that a binary code can correct more than $n/4$ errors only at the expenses of reducing the number of codeword to about $\log n$.

This is the route we pursued by deriving a binary code from a Reed-Solomon one; the latter is Maximum Distance Separable (MDS) and reaches the Singleton bound. The concatenation of the shortened Reed-Solomon code $[9600, 1920, 7681]_{2^{14}}$ and the $[14, 1, 1]_2$ mapping leads on average to a $[14 \times 9600, 1920, 7681]_2$ binary code. The correction rate is *de facto* increased only as we can decide which part of the codeword affect with errors and which not. And this is different than what happen in actual digital transmissions.

The idea is made clearer if instead of using a Reed-Solomon code, we generalize the construction to BCH codes. The software we employed for computing the iris code had $e_1 = 0.4$ and injecting errors in a restricted part of a longer codeword we might manage to use also this family of code. For example, let us use for the case at hand a $[32767, 2279, 7679]_2$ code that can correct up to $t = 3839$ errors. Performing $c \oplus I_1$ on the 9600 upper bit at enrollment and $s \oplus I_1'$ on the same substring at verification does not introduce any further error on the remaining $32767 - 9600$ bits. But now having gathered all the possible errors on a smaller part of the codeword, we also obtained a larger local correction ratio that is actually about $3839/9600 \approx 40\%$, as desired.

A second issue is that in the scheme described, the decoding procedure was successful when the number of different bits between the two iris codes was smaller than the error correcting capacity of the code. For Reed-Solomon codes, the classical Berkelekamp-Welch decoder can correct up to $t = \lceil \frac{n-k}{2} \rceil$ errors. But in [12] the authors showed that it is feasible to list all the codewords at a Hamming distance $t' > t$ (*list decoding* problem), with $t' \leq \lceil n - \sqrt{n(k-1)} - 1 \rceil$. Proceeding further in this direction, in [18] the authors managed to exploit the statistical characteristics of the channel and to solve the list decoding problem with even larger t'. While a larger number of errors corrected by an ECC decoder means more reliable transmissions and storage of information, here it implies that the user biometrics might be uncovered simply exploiting a more capable decoder. The solution is obvious: either a code for which list decoding algorithms are not available should be used, or the Reed-Solomon code should be tuned on the larger capacity decoder. The latter solution brings a wider computational burden (even if recent works show clear progress in reducing the computational time [11]).